Deep Irregular Convolutional Residual LSTM for Urban Traffic Passenger Flows Prediction

Bowen Du, Hao Peng, Senzhang Wang, Md Zakirul Alam Bhuiyan, Lihong Wang, Qiran Gong, Lin Liu, and Jing Li

***Abstract*— Urban traffic passenger flows prediction is practically important to facilitate many real applications including transportation management and public safety. Recently, deep learning based approaches are proposed to learn the spatio-temporal characteristics of the traffic passenger flows. However, it is still very challenging to handle some complex factors such as hybrid transportation lines, mixed traffic, transfer stations, and some extreme weathers. Considering the multi-channel and irregularity properties of urban traffic passenger flows in different transportation lines, a more efficient and fine-grained deep spatio-temporal feature learning model is necessary. In this paper, we propose a deep irregular convolutional residual LSTM network model called DST-ICRL for urban traffic passenger flows prediction. We first model the passenger flows among different traffic lines in a transportation network into multi-channel matrices analogous to the RGB pixel matrices of an image. Then, we propose a deep learning framework that integrates irregular convolutional residential network and LSTM units to learn the spatial-temporal feature representations. To fully utilize the historical passenger flows, we sample both the short-term and long-term historical traffic data, which can capture the periodicity and trend of the traffic passenger flows. In addition, we also fuse other external factors further to facilitate a real-time prediction. We conduct extensive experiments on different types of traffic passenger flows datasets including subway, taxi and bus flows in Beijing as well as bike flows in New York. The results show that the proposed DST-ICRL significantly outperforms both traditional and deep learning based urban traffic passenger flows prediction methods.**

***Index Terms*— Traffic passenger flows prediction, irregular convolutional neural network, LSTM, importance sampling, urban computing.**

Manuscript received February 2, 2018; revised September 28, 2018 and February 9, 2019; accepted February 16, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 51822802, Grant U1636210, Grant U1811463, Grant 51778033, and Grant 61772151, in part by the Beijing Advanced Innovation Center for Big Data and Brain Computing, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20171420, and in part by the Fordham University Faculty Startup Grant. The Associate Editor for this paper was K. Savla. *(Bowen Du, Hao Peng, and Senzhang Wang are co-first authors.) (Corresponding author: Hao Peng.)*

B. Du, H. Peng, Q. Gong, L. Liu, and J. Li are with the State Key Laboratory of Software Development Environment, Beihang University, Beijing 100083, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100083, China (e-mail: penghao@act.buaa.edu.cn).

S. Wang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China.

M. Z. A. Bhuiyan is with the Department of Computer and Information Sciences, Fordham University, New York, NY 10458 USA.

L. Wang is with the National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China.

Digital Object Identifier 10.1109/TITS.2019.2900481

I. Introduction

Accurately predicting the urban traffic passenger flow is of great importance for transportation resource scheduling, planning, public safety, and risk assessment [1]–[11]. With the development of urbanization and urban population expansion, accurately forecasting the inflow and outflow for each transport station are becoming more and more challenging. This is mainly because traffic passenger flows can be affected by multiple dynamic and complex factors including the dynamic traffic routes, the upgrading of transportation facilities, the complex transfer flows, effect of rush hours and other external factors such as bad weathers, etc [1], [12]–[15]. Most traditional approaches represent urban traffic passenger flows as two matrices with each one representing the region-level inflow or outflow data of a city, respectively. Although region-level traffic passenger flow model is a straightforward and efficient way to represent traffic passenger flows in a city, it ignores the independence as well as the interactions among different traffic lines. Thus they might not work well for line-level or station-level traffic passenger flows predictions.

Recently, deep learning has been proven to be effective to perform end-to-end learning of feature representations, and has made groundbreaking progress on object recondition in computer vision, speech recognition and text mining problems [16]. Some recent works also try to use deep learning models to capture the spatio-temporal traffic passenger flows features [1], [2], [9], [12], [17]–[20]. For example, [1], [12] proposed the popular ST-ResNet model which samples at a regular internals for *closeness*, *period*, *trend* and *external* influence on Residual Networks and gains a hopeful performance. However, there are two mainstream deep learning architectures which have attracted more research attention, i.e., recurrent neural networks (RNNs) [21], [22] and convolutional neural networks (CNNs) [23], [24] due to their powerful ability in handling spatial-temporal data. Although RNNs are especially powerful in modeling the sequential or temporal data [22], there still lacks of an effective RNNs based urban traffic passenger flows prediction model for traffic passenger flows representation learning due to the complexity of fine-grained spatial-temporal correlation learning among different traffic lines. As a simplified sequential learning model, the deep spatio-temporal residual networks (ST-ResNet) is proposed in [1] and [12]. However, mainstream models [1], [2], [9], [18]–[20], [25], [26] assume that there are only two channels

as the input, and thus it only works for the region-level inflow and outflow prediction. It did not consider the independence and interactions of different traffic lines and the continuity of traffic passenger flows. Different from RNNs, CNNs are more effective to capture the spatio correlations of the data through a convolutional mask to sequentially convolve over a matrix which can be the pixel matrix of an image. However, for a matrix constructed from the traffic passenger flow data, an irregular convolution kernel is necessary to transform the lower-level features to the higher-level features [27], [28]. Compared with regular CNNs, irregular CNNs may be much more effective to capture the irregular flows of the urban traffic lines in real scenarios, and thus they are more suitable to extract the complex and interpretable spatial features.

In this paper, we propose a Deep Spatio-Temporal traffic passenger flows feature learning model named DST-ICRL which combines the Irregular Convolutional Residual Network and the Long Short Term Memory (LSTM) Recurrent Neural Network for accurately predicting the urban traffic passenger flows. We first design a novel irregular convolutional residual neural network to learn the spatial traffic passenger flows features. For the temporal features learning, the effective LSTM model is applied to capture the near previous data, the periodicity and the trend of the traffic passenger flows data from the sampled short-term as well as long-term historical data. Specifically, the whole architecture of the proposed model is as follows.

Multi-Channel Input. Instead of viewing the urban traffic passenger inflow and outflow as two matrices corresponding to two channels, we propose to utilize semantical multi-channel matrices to represent the traffic passenger flows. A natural way to construct the matrices is based on the rail transit functions, i.e., urban subway routes flow. We consider each subway line as a channel and build two corresponding matrices to represent its inflow and outflow, respectively. For the urban bus passenger flows, we can factorize the entire public transportation network into several sub-networks by the functional areas, such as road grades or passenger flows, etc.

Convolution Layers. In order to effectively learn the spatial features in traffic passenger flows, we propose to utilize convolutional neural network with a carefully designed new irregular convolution kernel to capture the interpretable high-level dependency in a channel. As the traffic passenger flows patterns can be largely affected by the urban population distribution or urban functional area, we take advantage of those urban planning experiences to upgrade the interoperability of features. We use a deformable irregular convolution kernel to cover the neighborhood of each region and slide from left to right and from top to bottom. In terms of time acceleration efficiency, the parameters among irregular convolutional kernels are independent, so the proposed irregular CNNs can run on multiple CPUs or GPUs in parallel. Different from the image data which normally has three channels at most, i.e., RGB values, the traffic passenger flows may have much more channels with each one associated with a transport line. We let all the channels share the parameters, and thus we coordinately modify the configuration of all the following irregular convolution layers to make the feature representation more effective and efficient. In order to avoid the issue of gradient disappearing in model optimization, we also utilize the residual units [1].

Recurrent Layers. We next employ the popular Long Short Term Memory [29] (LSTM) framework to learn the temporal features of passenger flows and predict their future trend. We resize the multi-channel convolutional feature map into a sequence of features, and learn the spatio-temporal features with LSTM units. Previous study showed that the three time periods based sampling method including closeness, periodicity and trend are effective in modeling urban traffic passenger flows [1]. Following their work we also sample the three components of spatio-temporal data to predict the future traffic passenger flows. In addition, since the most recent traffic passenger flows of a region are highly correlated to the traffic passenger flows of the region in the next time slot, we sample the close samples with a larger probability than periodicity and trend's. It's an importance sampling instance when extracts training traffic flows with different probabilities.

Output. We use the multi-channel structures in training, and we sum all the output channel matrices into two channel matrices as the finally output layer. The advantages are that we can not only predict the traffic passenger flow for each station, but also for each route. The final inflow and outflow of a region can be obtained by summing the predicted values of all the channel matrices of the regions.

The main contributions of this paper are:

- We propose a novel deep irregular convolutional residual LSTM model named DST-ICRL to deeply capture the spatio-temporal traffic flow features for more accurately predicting urban traffic passenger flows. Different from previous deep models, DST-ICRL combines irregular CNN, importance sampling and residual networks to better fit the traffic passenger flow prediction task.
- For real urban traffic passenger flows, we demonstrate that multi-channel modeling, irregular convolutional kernel and important sampling are more suitable to represent and predict the traffic passenger flows data. This can be a general framework for deep learning model to be applied in modeling traffic passenger flows data.
- We compare DST-ICRL with state-of-the-art deep learning approaches on four benchmark datasets which contain different types of traffic passenger flows for evaluation. The results show that DST-ICRL significantly outperforms all the baselines in both effectiveness and efficiency in urban traffic passenger flows prediction.

The rest of the paper is organized as follows. We first review related work in Section II. Problem definition and data preprocessing is given in Section III. The details of the framework is introduced in Sections III and IV. Section V evaluates the model performance, followed by the conclusion of this work in Section VI. The code of this work is publicly available at https://github.com/RingBDStack/Deep-Convolutional-Residual-LSTM.

II. Related Work

In this section, we will briefly review the related work. Traffic passenger flows prediction models can be roughly

categorized into traditional shallow prediction models and deep learning based prediction models. Next we review the related work in the following two categories.

A. Traditional Traffic Passenger Flows Prediction Models

Traffic prediction problems mostly focus on short-term prediction and long-term prediction tasks [30]. Tradition traffic passenger flows predictions rely on feature engineering and selection to obtain good features for prediction task. Generally, traditional traffic passenger flows prediction approaches can be categorized into parametric methods, including Auto-Regressive Integrated Moving Average (ARIMA) based methods [31]–[34] and non-parametric methods, including K-nearest neighbor (KNN) nonparametric regression methods, historical average(HA), vector Autoregressive (VAR), gaussian process based [35], [36], etc. However, ARIMA-based models are not suitable for analyzing time series with missing data, since they rely on uninterrupted time series data. HA model cannot effectively capture dynamic changes of the traffic data, such as incidents or social events. VAR model can capture the linear inter-dependencies among inter-related time series, but the correlation between the predicted values is neglected. In addition to parametric and non-parametric based predictive models, there are some researchers have attempted to combine multi-source traffic data to handle external factors such as traffic accidents [37]–[40] and weather [41]. Traditional works are different from ours where the above proposed methods naturally focus on predicting the traffic passenger flows of one particular region, such as a street or a local region, and they do not conduct the city-level traffic passenger flows prediction.

B. Deep Learning Based Traffic Passenger Flows Prediction Models

With the growing popularity of deep learning techniques and the success of various deep learning algorithms in many fields such as pattern recognition and natural language processing, recently some works also try to apply deep learning models in traffic prediction and have achieved promising results. Existing deep learning models [1], [2], [2], [8]–[11], [13], [18]–[20], [25], [42] are inspired from image recognition and model the urban traffic passenger flows as matrices like the gray-level matrices of the images. In the spatio-temporal flow features extraction layers, the traditional efficient deep convolution neural networks [43]–[45] and deep residual convolution neural networks [46] models have gained good performance. The residual framework has shown powerful spatio-temporal feature extraction ability [1], [12]. In addition to traffic passenger flows, spatio-temporal based deep learning models have also been applied in urban traffic speed prediction tasks. Recent deep learning based traffic flows prediction models, such as DeepST [47], ST-ResNet [1], AttConvLSTM [48], DMVST-Net [26], and DCRNN [25], have achieved the state-of-art performances. ST-ResNet [1] introduces four major components to model the temporal closeness, period, trend and external information, and utilizes the residual neural network to predict the crowd flows of a city. Although ST-ResNet incorporates prior knowledge to sample the input historical traffic flow data in different time periods for forecasting the future traffic flows, it still lacks the analysis of the independence among different traffic lines. AttConvLSTM [48] employs convolutional LSTM units and utilizes the attention mechanism to emphasize the effects of representative citywide demand patterns on each-step prediction. However, it ignores the periodicity of the traffic flow data. DMVST-Net [26] employs local CNN, LSTM and semantic graph embedding to integrate the spatial, temporal, and semantic views, respectively. DCRNN [25] captures the spatial dependency by using bidirectional random walks on the road network and the temporal dependency by using the encoder-decoder architecture with scheduled sampling. However, existing models [1], [13], [25], [26] simply modeled traffic passenger flows into 1 or 2 layers matrices, and lacked superior and recursive time-series model for the large computation in traffic passenger flows matrices. Compared to the proposed DST-ICLR model, the DCRNN [25] model cannot capture the different attributes of traffic passenger flows, such different lines and different functional areas, and ignore the modeling of different weights of historical data due to only using RNN unit.

III. Problem Definition and Data Preprocessing

In this section, we will first define the studied problem. Then we will introduce the traffic passenger flow data studied in this paper. Next, we will give a brief introduction on how to decompose traffic passenger flows into multiple-channel matrixes, and apply convolutional operators in the multiple channels.

Definition 1 *Cell Region. In this study, we partition a city into an $I \times J$ grid map based on the longitude and latitude. Each grid is defined as a cell region, and all the grids form a cell region set $R = \{r_{1,1}, \cdots, r_{i,j}, \cdots, r_{I,J}\}$, where $r_{i,j}$ is the cell region in the i-th row, j-th column of the grid map.*

Definition 2 *Inflow/Outflow [1].* Given a collection of crowd flow trajectories $\mathcal{Q}$, for a cell region $r_{i,j}$ the inflow and outflow of the crowds at time t are defined as follows respectively,

$$X^t_{in,i,j} = \sum_{T_r \in \mathcal{Q}} |\{k > 1 | g_{k-1} \notin r_{i,j} \wedge g_k \in r_{i,j}\}|, \quad (1)$$

$$X^t_{out,i,j} = \sum_{T_r \in \mathcal{Q}} |\{k \geq 1 | g_k \in r_{i,j} \wedge g_{k+1} \notin r_{i,j}\}|, \quad (2)$$

where $T_r : g_1 \rightarrow g_2 \rightarrow \cdots \rightarrow g_{T_r}$ is a trajectory in $\mathcal{Q}$, and g_k is the geo-spatial coordinate.

Following previous works [1], [49], we also denote the inflow and outflow in all the cell regions in time slot t as a crowd flow tensor $\widehat{X}^t \in \mathcal{R}^{i \times j \times d}$. Based on the above definitions, we formally define the studied problem as follows.

Problem Definition Traffic Flow Prediction *Given the traffic passenger flow tensors $\{\widehat{X}^t | t = 1, 2, ...n\}$ in the cell regions R over the previous n time slots, out goal is to predict the traffic passenger flow tensor $\widehat{X}^{n+1}$ for the next time slot $n + 1$.*

In this study, the raw data is the trip records of the anonymous passengers. Each record contains the traffic route number/train number, the entrance station ID and time, the exit station ID and time and the location (longitude, latitude). The traffic flow in a region $r_{i,j}$ in time slot t is the sum of all the check-in or check-out passengers. Similar to the

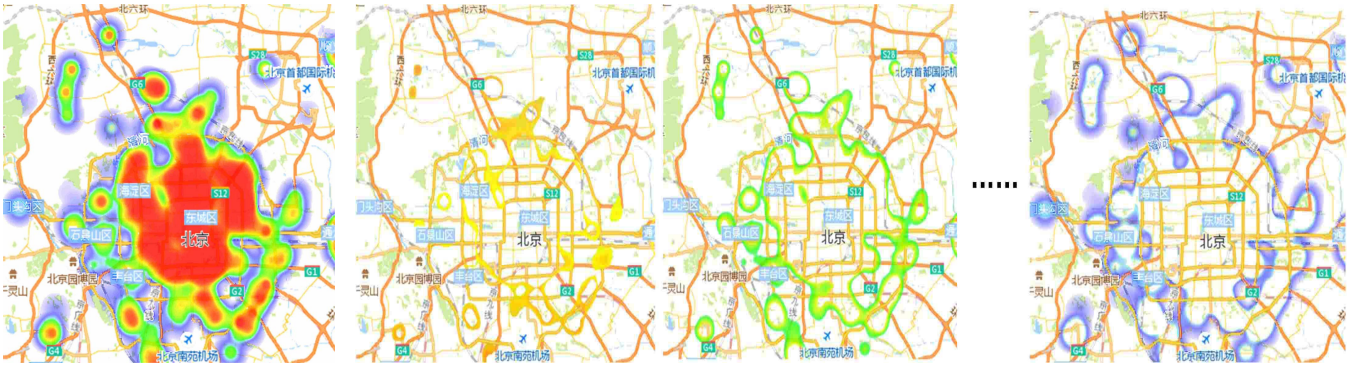


Fig. 1. Multiple channel matrices. The left is an original single channel flows heat matrix. Different colors represent different traffic passenger flows volume, and the green color means a small traffic passenger flows volume while the red color means a large traffic passenger flows volume. The right three maps are the decomposed multi-channel representations of the traffic passenger flows. (a) Original map. (b) Multi-channel decomposition map.

RGB channels in an image, we decompose the urban transport lines into multiple channels. We take the subway lines as an example to show how to construct multi-channel matrices to model traffic flows. Since the traffic of passengers on different traffic lines or routes is mostly independent, except for some transfer stations where passengers of different traffic lines can transfer from a line to another. We divide the passenger flows of different traffic lines or routes into different channels. In addition, the traffic passenger flows in each line or route can be further decomposed into two channels which correspond to inflow and outflow, respectively. There are three advantages of decomposing traffic passenger flows into multiple channels. Firstly, it's relatively easy to expand the spatial representations when a new traffic route emerged. Secondly, it can reduce the complexity of modeling the inflow and outflow of multiple merged transport lines. Thirdly, we use multiple channels to ensure the prediction on a traffic line is not affected by the traffic of other lines. Given the passenger flow trajectories Q, we further decompose it into D channels with each channel corresponding to a matrix q of the size $I \times J$ as follows,

$$X^t_{in,i,j} = \sum_{d=1}^{D/2} x^t_{in,d,i,j} \tag{3}$$

$$X^t_{out,i,j} = \sum_{d=D/2}^{D} x^t_{out,d,i,j} \tag{4}$$

where $x^t_{in,d,i,j}$ and $x^t_{out,d,i,j}$ represent the inflow and outflow of the passengers of the region $r_{i,j}$, the d refers to channel, and the time slot is t, respectively. For example, as shown in Figure 1, we decompose the original traffic passenger flows of Beijing subway into 36-channel representations. Since the Beijing subway has 18 lines and the traffic passenger flows of these lines can be considered independently, we decompose the subway flows into 36 channels by lines.

When CNN is applied to image data, a fixed size convolution mask (e.g., 11×11 pixels used in AlexNet [44]) is applied to the local patches on the image to extract low-level features, e.g., edges. The combinations of the convolved features are further convolved in the next layer to obtain higher-level feature representations, e.g., parts and objects. By analogy to images, we also try to apply convolution masks to the urban traffic passenger flows in different regions of a city to learn high level features. The traffic passenger flows of the entire city can be considered as an image, and the traffic passenger flows in a region of the city can be considered as a pixel. In this study, we partition a city into a $I \times J$ grid regions as shown in Figure 2(a). Here we follow the multi-channel representation approach in Eq. (3) and (4), and the original traffic passenger flows matrix is composite of multi-channel matrices. To further illustrate the multi-channel traffic passenger flows, we give a visualization of the decomposed multiple channels of traffic passenger flows of Beijing subway in Figure 2(b). The sum of the values on the corresponding regions on each channel matrix is equivalent to the value in the region of the original matrix. In order to better map the real urban traffic passenger flows, we use multiple pixels to represent a traffic site in each channel matrix. Then we can perform the general deep convolutional neural networks in multi-channel flows c to extract high-level spatial features.

IV. Irregular Convolution Based Residual LSTM Model

In this section, we introduce the proposed irregular convolution based residual LSTM model in details. The architecture of the proposed model is shown in Figure 3. The model contains four major parts, the residual convolutional layers, the LSTM layers, importance sampling and the model fusion component. We also see that the input data are first processed by the irregular convolutional filters. The details will be illustrated in section IV-A. Then, parameter shared convolutional module is presented to learn temporal and spatial features among multiple channels of the traffic passenger flows. Next, the global architecture integrates the residual neural network and LSTM units to learn the high-level temporal and spatial features. Finally, the data fusion component combines the predictions based on the previous neighborhood data, the periodicity, the trend, as well as the external factors such as weather, the day of the week, traffic control, sports event and vocal concert, etc. to make a final prediction.

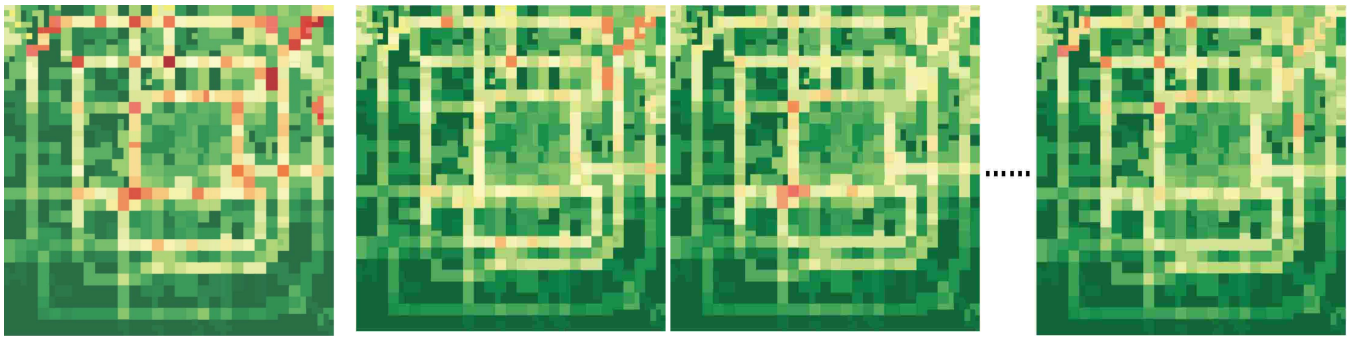


Fig. 2. Multi-channel matrix representation of the traffic passenger flows data. The left is an original one-channel traffic passenger flows heat matrix. Different colors represent different traffic passenger flows volume, and the green color means a small traffic passenger flows volume while the red color means a large traffic passenger flows volume. The right matrices are the decomposed multi-channel matrix representations of the traffic passenger flows. (a) Single channel flows matrix. (b) Multiple channel flows matrices.

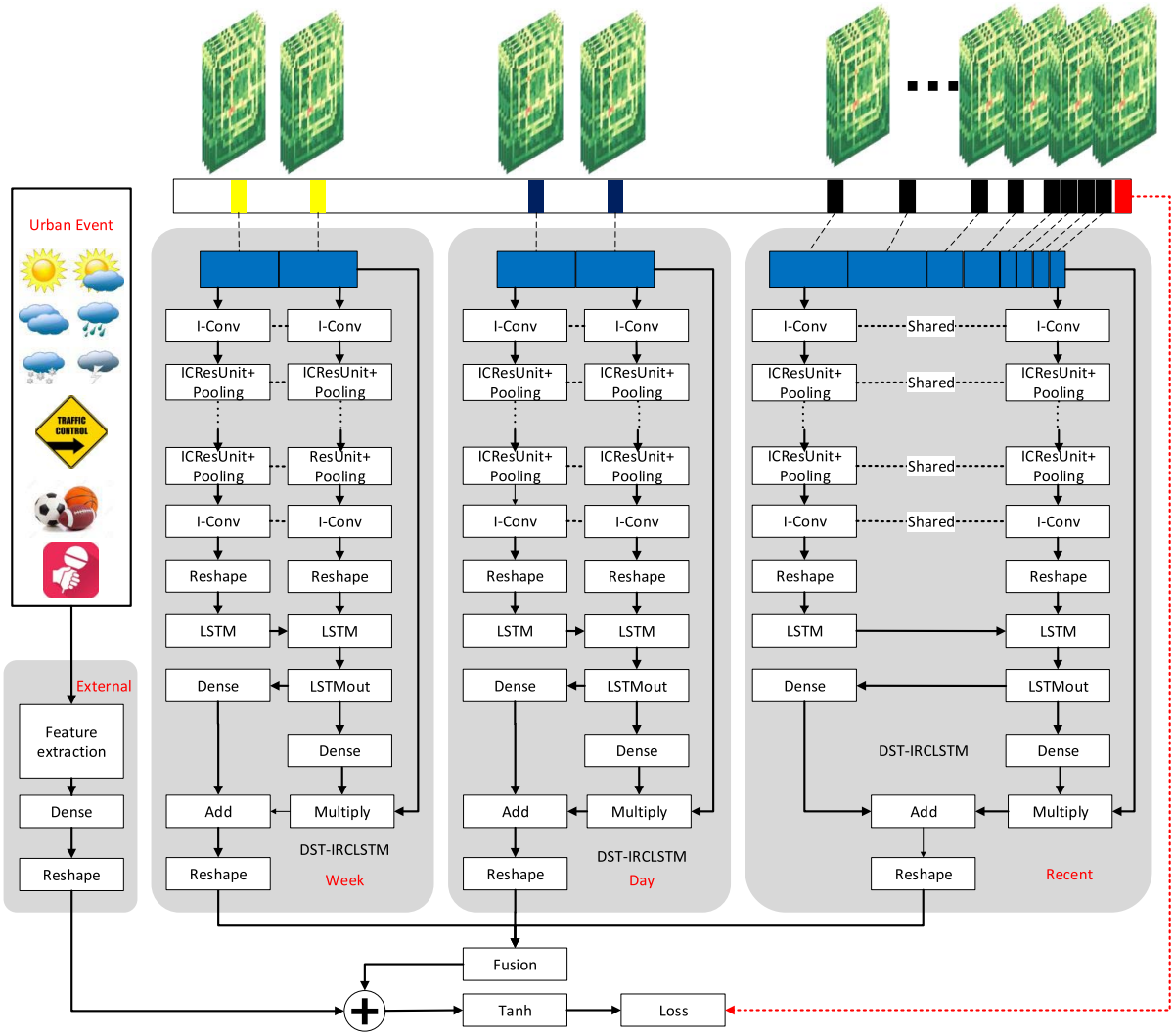
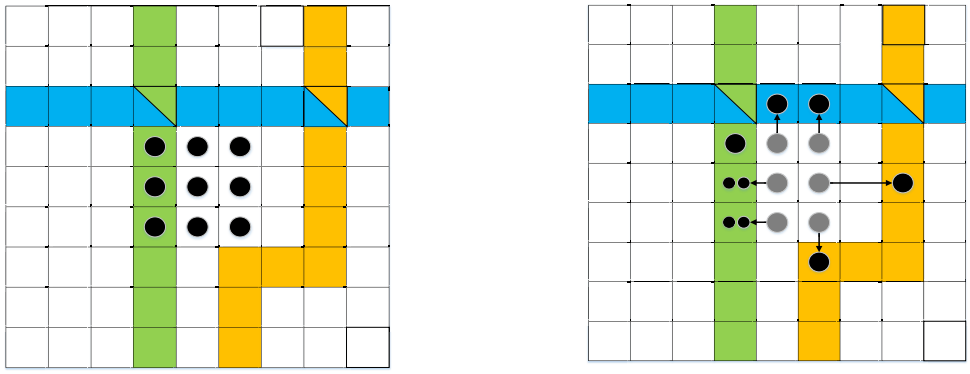


Fig. 3. The framework of the proposed DST-ICRL. I-Conv: Irregular Convolution; ICResUnit+Pooling: Irregular Convolution Residual Neral Network Units and Max Pooling; LSTMout: the output of LSTM.

A. Irregular Convolution Neural Networks

A commonly used convolution kernel in image processing is a small matrix, which is not the best kernel in our case. In this paper, we propose an irregular convolution to fit the urban traffic passenger flows matrix. In traffic passenger flows prediction, only the regions with transport lines have the inflow



(a) (b)

Fig. 4. Irregular Convolution. There are three transport lines marked with yellow, green and blue colors respectively, and a 3×3 convolutional kernel. The crossing pixels represent the transfer stations. The left is the traditional square 3×3 convolution kernel. The right is our irregular convolution. In the convolutional mask, as there are no transport lines in regions with the gray circles, our irregular convolution kernel chooses the near station values to filter features. The small black circles represent repeated sampling pixels. (a) Illustration of the original convolution. (b) Illustration of the proposed irregular convolution.

and outflow data, while there is no traffic passenger flows in the regions without any transport lines passing over. Intuitively, the flows in traffic site are affected by the transport routes and the nearby sites, which correspond to the inflow and outflow station and the the transfer station. Thus the widely used square convolution kernel does not fit the traffic passenger flows data. To effectively extract traffic features we need a new irregular shape of convolution kernel which can capture the property of the traffic passenger flows correlations among regions. As shown in Figure 4, the traditional convolution not only covers the pixels with traffic passenger flows, but also covers more hollow points. As most of traffic passenger flows are along the transport lines, the regions containing traffic lines can reflect the traffic passenger flows features while the regions without transport lines are much less helpful to predict the traffic passenger flows. When the convolution processes hollow pixel, instead of convolving the hollow region itself, we choose to convolve the nearest region to the hollow region that contains transport line. The irregular convolution still maintains the independence of the operations among convolutional kernels, thus it can be adapted to multi-threaded CPU or GPU computing architectures.

Because we sample a batch of historical traffic passenger flows data to train the model for predicting the future trend, as shown in Figure 3, both batch of samples and multi-channel flows representations bring in the problem of how to share the parameters among matrices. Similar to the RGB images or multi-channel word embeddings, we adopt a parameters sharing strategy. As shown in Figure 5, when we sample 8 flow matrices, and the size of each matrix is $128 \times 128 \times 36$. Firstly, we reshape the input streaming traffic passenger flows data as the standard multi-channel traffic passenger flows matrices, such as the 8 independent slices by time-series. Then, we make use of the irregular convolution and residual neural network to learn spatial and temporal features from the multi-channel matrices. At last, we also bring in the residual convolution neural network architectures to avoid gradient disappearance and learn high-level features.

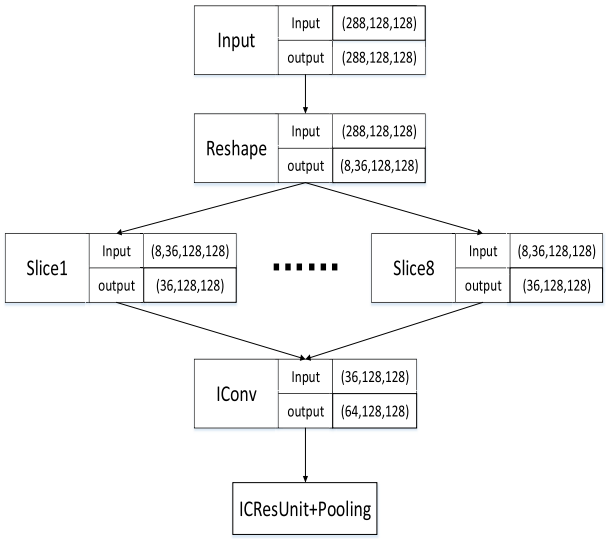


Fig. 5. The structure of the parameters shared convolution network.

B. Residual LSTM Learning

The residual convolutional neural networks have been proven to be very effective in high-level discriminative features learning from image and traffic passenger flows data. A major advantage of the residual convolutional neural networks is that it can build very deep convolution networks with hundreds of hidden layers. Due to its powerful feature learning ability, in this paper, we also take advantage of this framework. To make it fit our studied problems, we replace the traditional convolution kernel to the proposed irregular convolution kernel in the residual unites named ICResUnit, as shown in Figure 6. The more comprehensive illustration is shown in Figure 3, the input of the first layer ICResUnit is the output of the irregular convolution layer, namely I-Conv for short. For all forecasting tasks, we use 5 layers of ICResUnit, and each ICResUnit is followed by a max pooling operation.

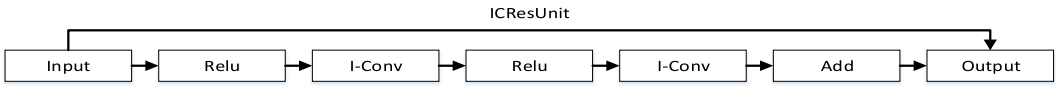


Fig. 6. The structure of the irregular convolutional residual neural network.

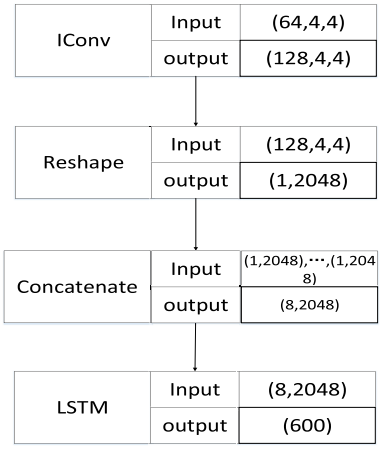


Fig. 7. The structure and parameters configuration of the LSTM unit.

To learn the temporal features of traffic passenger flows, we design an adaptive LSTM learning unit. As shown in Figure 7, we first reshape the $128 \times 4 \times 4$ feature representation into a 1×2048 feature vector. Next, we partition the feature vector into 8 smaller vectors of the same size with each one associated to a sampled historical traffic passenger flows snapshot. Then we sequentially concatenate the 8 feature vectors as the input of the LSTM unit. The recurrent neural network learning is shown in Figure 3, we can see that the output of the learned latent feature vector of the traffic passenger flows data in the current time frame is the input of the next layer of the LSTM model named LSTMout. The result of the dense operation is 2 channels feature map. To avoid the gradient disappearance, we design multiple modules between the feature mapping from LSTMout to the original flows. We sum all the predicted flows in each channel to the final total 2-channel inflow and outflow. More specifically, we add the results of multiplication and previous dense operation. Notice that the timing of LSTM is consistent with the real-time and sampling direction. Next, for fusing different components feature map, we reshape the LSTM result to multiple-channel matrix representations.

C. Features Fusion

As shown in Figure 3, the proposed DST-ICRL model is comprised of four major components modeling the temporal traffic passenger flows data including *recent*, *day*, *week* and the *external* influence factors, respectively. We first sample enough inflow and outflow data throughout a city at each time interval and model them as multi-channel image-like matrices. For the short-term historic traffic passenger flows data, we sample 8 snapshots of the traffic passenger flows data matrix in different time intervals. In this paper, the sampled recent data are consisted of 2 traffic passenger flows samples in the last 5 minutes, 2 samples in the last 10 minutes flows, 2 samples in the last 20 minutes and 2 samples in the last 30 minutes. For the daily historical traffic data part, we sample 2 samples of previous days in the same time interval. Similarity, the week component shares the same architecture with the day component. After irregular convolutional networks and ICResUnit and LSTM layers of features extraction, we adopt a multi-channel features fusion method. We finally fuse the learned spatial-temporal feature vectors of the above three parts of historical data as an unified spatial-temporal feature vector as follows,

$$X_{Fusion} = W_r \circ X_{Recent} + W_d \circ X_{Day} + W_w \circ X_{Week} \quad (5)$$

where the X_{Fusion} represents the fused unified prediction results, and X_{Recent}, X_{Day} and X_{Week} are the prediction results based on the three parts of sampled historical data, respectively. W_r, W_d and W_w are the parameter matrices. The $\circ$ represents the multiplication of the corresponding coordinate of values of the features and corresponding weights.

As some external factors can also have significant impact on the traffic passenger flows prediction, we also incorporate some external factors including weather situations, regular traffic control and various social events happened in the urban area, etc to further improve the prediction accuracy. Note that, we only choose the events that can cause a global impact on the entire urban area like bad weather, and we take no account of the events with local effectiveness, such as road accident, traffic jam and social events, etc. We encode the influence of the external factors into the multi-channel matrix which shares the same dimension with the traffic passenger flows feature matrix X_{Fusion}. In practice, the entry value in the urban event influence matrix should reflect the impact of the event on urban traffic passenger flows. Similar to flows matrix, we assign the value in coordinates which corresponds to specific transit stations by statistical the external influences and flows. A larger value in the event matrix means a greater effect of it on the traffic passenger flows. Here we also normalize the impact matrix X_{Ext} to the range -1 to 1. By encoding the external factors, the final prediction result can be obtained by a $tanh$ function fusing two parts of data as following,

$$\widehat{X}_{n+1} = tanh(W_f \circ X_{Fusion} + W_x \circ X_{Ext}) \quad (6)$$

where $\widehat{X}_{n+1}$ is the prediction of the inflow and outflow of traffic in the next time slot, and W_f and W_x are the parameter matrices. The predicted flows are the output of the $tanh$ activation function ranging from -1 to 1.

The input of DST-ICRL model is multi-channel passenger flow matrices, and the output is also a multi-channel matrices of the same size. Each channel output matrix is the passenger flow prediction for the corresponding traffic line. The total number of layers in our DST-ICRL model is 30, including 12 layers of irregular convolutional neural networks, 1 layer

TABLE I
DATASETS STATISTICS

Dataset	SubwayBJ	BusBJ	TaxiBJ	BikeNYC
Data type	e-card	e-card	Taxi GPS	Bike rent
Location	Beijing	Beijing	Beijing	New York
Time Span	7/1/2016-30/10/2016	7/1/2016-30/10/2016	7/1/2013-30/10/2016	4/1/2014-9/30/2014
Data frequency	15 minutes	15 minutes	30 minutes	1 hour
Grid region size	(128,128)	(128,128)	(32,32)	(16,8)

of LSTM unit and 1 layer of feature fusion. We employ a temporal min-batch samples to train the DST-ICRL model. The input includes 12 group of previous traffic passenger flows samples, and the output is the traffic passenger flows in the next time slice, as shown in Figure 3. The min-batch size is 32. We use the RMSE and MAE to measure the loss between the predicted values and ground truth flows. We calculate the accumulated loss for each region in min-batch and average all the loss values for all the regions as the final prediction loss. To speed up training, we choose the momentum [50] method on multiple GPUs.

V. EXPERIMENT

In this section, we conduct extensive experiments to evaluate the proposed models. We will first introduce the four different traffic passenger flows datasets, including subway, bus, taxi and bike, used in this work, and the state-of-the-art baselines for comparison. Then we will give the experimental settings including the default model parameters and the experiment environments. Next we will conduct quantitative evaluation of various models over the four datasets, and show the experimental results. To demonstrate the efficiency of our model, we also show the running time of the proposed models.

A. Datasets

We choose the following four datasets for evaluation, including the Beijing subway dataset, Beijing bus dataset, Beijing taxi dataset and New York City bike dataset. We can see that these four datasets represent different types of transportation modes. A summarization of the statistics of these four datasets is shown in Table I.

- **Beijing Subway (SubwayBJ)**: The Beijing subway data is collected from people's check-in and check-out records of their Beijing metro-card. The time span of this data is from 1st Jul.2016 to 30th Oct. 2016. Beijing subway has 18 lines, and the recording interval is 15 minutes. For this work, we obtain two types of crowd flows, and 2×18 channels of matrix representations. We construct a 128×128 grid regions of Beijing for this dataset. The data in the first three and a half months are used for training, and the remaining data are used for testing.
- **Beijing Bus (BusBJ)**: The Beijing Bus data is collected from people's getting on and getting off records of Beijing buses by their bus e-cards. The time span is from 1st Jul.2016 to 30th Oct. 2016. Beijing bus has 1020 lines, and the recording interval is 15 minutes. We obtain two types of crowd flows, and $2\times$N channels of representations. We also construct a 128×128 grid regions for this dataset. The first two months data are used for training, and the remaining one month data are used for testing.
- **Beijing Taxi (TaxiBJ)**: The Beijing Taxi data is collected from the taxicab GPS data in Beijing in four time intervals: 1st Jul.2013 - 30th Oct. 2013, 1st Mar. 2014 - 30th Jun. 2014, 1st Mar. 2015 - 30th Jun. 2015, 1st Nov. 2015 - 10th Apr.2016. We map the traffic passenger flows of this data into 2 channels and the grid size is 32×32. The data of the last four weeks are the testing data, and the other data are training data.
- **NewYork Bike (BikeNYC)**: This data is released by the NYC Bike system in 2014, from Apr. 1st to Sept. 30th. Each trip data includes: trip duration, starting and ending station IDs, and start and end times. We use the last 10 days data for testing and the remaining data are used for training.

B. Baselines and Benchmark

We compare DST-ICRL with the following baselines.

- **Historial Average (HA)**: It simply uses the historical average of the same time period and same region as the prediction. For example, to predict the traffic passenger flows of region r in 9:00am-9:30am, we use the average traffic passenger flows of region r in all the previous days in the same time interval 9:00am-9:30am as the prediction.
- **Auto-Regressive Integrated Moving Average (ARIMA)**: It is a well-known model for understanding and predicting the future trends of time series data, and widely used in traffic flow prediction.
- **SARIMA**: It is a seasonal ARIMA model, and considers the seasonal terms, capable of both learning closeness and periodic dependencies beyond ARIMA.
- **Vector Auto-Regressive (VAR)**: It captures the pairwise relationships among all flows, which is an advanced spatio-temporal model and has heavy computational costs due to the large number of parameters.
- **ST-ANN**: It extracts spatial (nearby 8 regionsâĽ™ values) and temporal (8 previous time intervals) correlated traffic passenger flows data as the input, and then they are fed into an artificial neural network.
- **DeepST** [47]: It is a deep neural network based prediction model, and models the spatial-temporal data as temporal closeness, period and seasonal trend. This model shows state-of-the-art results on the crowd flow prediction.
- **ST-ResNet** [1]: It is currently the state-of-the-art deep convolution-based residual networks for predicting the future urban traffic passenger flows [1]. The major difference between this model with ours is that ST-ResNet only has two

channel data as the input, and it uses the traditional square convolution kernel.

• **AttConvLSTM** [48]: It employs an encoder-decoder framework based on convolutional and attentional LSTM to capture the spatial-temporal features. It is a state-of-the-art model for multi-step passenger demands prediction in the mobility-on-demand services.

• **DMVST-Net** [26]: DMVST-Net is a deep multi-view spatial-temporal neural network model for taxi demand prediction. It incorporates information of the following three views: temporal view, spatial view and semantic view.

• **DCRNN** [25]: DCRNN is a diffusion convolutional recurrent neural network based model for traffic forecasting. It uses bidirectional graph random walk to model the spatial dependency and recurrent neural network to capture the temporal dynamics.

We use both the Rooted Mean Square Error (RMSE) and the Mean Average Error (MAE) as the evaluation metrics,

$$MAE = \frac{1}{z}\sum_{i} ||x_i - \hat{x}_i||, \tag{7}$$

$$RMSE = \sqrt{\frac{1}{z}\sum_{i}(x_i - \hat{x}_i)^2}, \tag{8}$$

where $\hat{x}$ and x are the predicted value and the ground truth, respectively, and z is the number of all the samples for prediction. As the traffic passenger flows contains inflow and outflow, we also use $RMSE_{in}$ and $RMSE_{out}$ to denote the $RMSE$ of the inflow and outflow respectively.

C. Experimental Settings

For the input of the proposed DST-ICRL, we use different numbers of channels to represent the initial input traffic passenger flows data for different datasets. For example, the Beijing subway has 18 lines and thus we model it as 36 channel matrices. Each channel matrix is with the size of 128×128. The bus traffic has 40 channel matrix with 128×128, the taxi traffic has 2 channel matrix with 32×32 and the bike traffic has 2 channel matrix with 16×8. We first use the Min-Max normalization method to scale the input data into the range [0,1]. In the evaluation, we re-scale the predicted value back to the normal values. For external factors, we also adopt one hot coding to transform the meta-data including weather, weekend, weekday, holiday, traffic control, sport events and vocal concert into binary vectors, and use Min-Max normalization to scale the external influences into the range [0,1].

All of our experiments were performed on 64 core Intel Xeon CPU E5-2680 v4@2.40GHz with 512GB RAM and 4×NVIDIA Tesla P100-PICE GPUs. The operating system and software platforms are Debian 7.0, TensorFlow r0.12 and Python 3.4. The convolutions of I-Conv and all the residual units use 128 filters of size 3×3. The batch size is 128 in all the experiments. For the AttConvLSTM model, the number of nodes in the single hidden layer of MLP is set to 1024, the batch size in mini-batch optimization is set to 16, and the learning rate in the Adam optimizer is set to 0.0002. The parameters of the other baseline methods are set following the setting in their original papers. We also use early-stop in all the experiments.

TABLE II
COMPARISON AMONG DIFFERENT METHODS ON SUBWAYBJ

Model	RMSE	MAE
HA	0.0245	0.0077
ARIMA	0.0193	0.0060
SARIMA	0.0217	0.0071
VAR	0.0174	0.0057
ST-ANN	0.0142	0.0043
DeepST	0.0113	0.0034
AttConvLSTM	0.0081	0.0024
ST-ResNet	0.0079	0.0023
DMVST-Net	0.0072	0.0020
DCRNN	0.0065	0.0013
DST-ICRL	**0.0007**	**0.0002**

D. Experiment Results on SubwayBJ

RMSE and MAE values of these methods over the Beijing subway dataset are shown in Table II. We can observe that the proposed DST-ICRL model significantly outperforms all the baselines. Comparing with the previous state-of-the-art models, DST-ICRL achieves the lowest RMSE 0.0007 and the lowest MAE 0.0002 among all the methods, and the performance improvements are both significant on the two metrics.

Considering traditional baselines based on the shallow traffic passenger flows representation, we also perform the same 2-channel grids on our DST-ICRL model, which reduces the error of outflow into 0.0067. Under the 2-channel constraints, DST-ICRL achieves similar performance as the strong baseline DCRNN. From this result, one can see that the multi-channel representations are much more suitable for high-level traffic feature learning compared to the two-channel representation. We also give the results of 3 variants of DST-ICRL with different settings in Table III. For regular convolution, the average RMSE is 0.0011, which indicates that our irregular convolution filtering is effective. For the importance sampling strategy, when we perform the uniform sampling in recent part, the average RMSE is 0.0011. It also indicates that our importance sampling method is an effective way to select the training data. In addition, we also give the predictions for the inflow and outflow traffics, respectively. The DST-ICRL reduces the error of outflow to 0.00068, inflow to 0.00088. In summary, the DST-ICRL model achieves the best results in the subway traffic forecasting.

E. Experiment Results on BusBJ

Table IV shows the results of our model and baseline methods on the Beijing bus dataset. The Beijing bus has a much more complex transport line distribution compared to the Beijing Subway. In total, in Beijing there are around 1040 bus lines and 29525 buses. It is not suitable to represent each Beijing bus line as a channel because that would lead to

TABLE III
COMPARISON AMONG DIFFERENT SETTINGS OF DST-ICRL ON SUBWAYBJ

Differential Setting	RMSE	RMSE-IN	RMSE-OUT
2 channel	0.0067	0.0069	0.0065
uniform sampling	0.00094	0.0009	0.00088
regular convolution	0.0011	0.0012	0.0010
none	**0.0007**	**0.00088**	**0.00068**

TABLE IV
COMPARISON AMONG DIFFERENT METHODS ON BUSBJ

Model	RMSE	MAE
HA	0.0082	0.0026
ARIMA	0.0067	0.0022
SARIMA	0.0088	0.0028
VAR	0.0041	0.0012
ST-ANN	0.0034	0.0011
DeepST	0.0028	0.00075
AttConvLSTM	0.0025	0.00072
ST-ResNet	0.0022	0.00070
DMVST-Net	0.0018	0.00057
DCRNN	0.0017	0.00055
DST-ICRL(2 channel)	0.0015	0.00048
DST-ICRL(uniform sampling)	0.00088	0.00028
DST-ICRL (importance sampling)	**0.00047**	**0.00014**

too many channels. For the sake of simplicity, we randomly sample 26 bus lines to form one channel, so in total we have 40 channels grid region representation for the Beijing bus traffic passenger flows. From table IV one can see that the RMSE and MAE of DST-ICRL is 0.00088 and 0.00028 respectively by uniform sampling the training samples. When we use the importance sampling method, the RMSE is further reduced to 0.00047, and MAE is reduced to 0.00014. For a fair comparison with traditional models, we can see that if only 2-channel (inflow and outflow) matrix inputs are used, the RMSE of DST-ICRL is 0.0015 and inferior to DST-ICRL with 40 channels, but outperforms the other baseline methods. Among all the baselines, DCRNN achieves the best performance. RMSE and MAE of DCRNN are 0.0017 and 0.00055, respectively. Our method significantly outperforms all the baseline methods. It shows again that the irregular convolutional neural networks and LSTM models can better improve the performance of extracting spatio-temporal features than traditional convolution kernel based models. Generally, previous deep learning models including DeepST, AttConvLSTM, ST-ResNet, DMVST-Net and DCRNN perform better than traditional shallow methods such as HA, ARIMA, SARIMA, VAR, and ST-ANN.

F. Experiment Results on TaxiBJ

Table V shows the experiment results of various methods on the Bejing taxi dataset. Different from subway and bus traffic passenger flows having fixed routes, it's very difficult to divide taxi flows into specific channels. The taxi route is instantly determined by passenger and traffic environments, such as congestion status, passenger's preferences and limited

TABLE V
COMPARISON AMONG DIFFERENT METHODS ON TAXIBJ

Model	RMSE	MAE
HA	57.69	18.91
ARIMA	22.78	7.25
SARIMA	26.88	8.51
VAR	22.88	7.47
ST-ANN	19.57	6.23
DeepST	18.18	6.21
AttConvLSTM	17.41	6.04
ST-ResNet	16.69	5.41
DMVST-Net	15.57	5.28
DCRNN	15.04	5.10
DST-ICRL (uniform sampling)	14.77	4.77
DST-ICRL (importance sampling)	**14.07**	**4.62**

TABLE VI
COMPARISON AMONG DIFFERENT METHODS ON BIKENYC

Model	RMSE	MAE
ARIMA	10.07	6.41
SARIMA	10.56	5.44
VAR	9.92	6.33
DeepST	7.43	4.25
AttConvLSTM	7.09	4.19
ST-ResNet	6.33	4.03
DMVST-Net	6.01	3.95
DCRNN 5.97	3.88	
DST-ICRL (uniform sampling)	5.93	3.11
DST-ICRL (importance sampling)	**5.77**	**3.05**

choices of traffic lines, etc. Moreover, the taxi routes are more likely to be affected by external factors such as weather. For a fair comparison, we use 2-channel (inflow and outflow) input data for all the deep models. We can see that the proposed DST-ICRL model still achieves the best performance with the smallest RMSE value 14.07 and MAE value 4.62. Because of the uncertainty and the randomness of the taxi data, the traditional regressive and average based models cannot achieve satisfied performance. The result shows that irregular convolution based residual LSTM and important sampling approaches can reduce RMSE by 0.97 compared with DCRNN. Compared with the uniform sampling for training taxi flows, the important sampling strategy can reduce RMSE by around 0.7 and MAE by around 0.15. Compared with DCRNN, which obtained the highest prediction accuracy in TaxiBJ dataset among all the baseline models, DST-ICRL decreases RMSE by about 1 and MAE by 0.48.

G. Experiment Results on BikeNYC

Table VI shows the results of our model and other baselines on BikeNYC dataset. BikeNYC consists of two different types of crowd flows, including new-flow and end-flow. Similar to Taxi flow, the BikeNYC flow is also different to be divided into multi-channel representations. For a fair comparison, we also use 2-channel (inflow and outflow) input data for all the deep models. We can observe that the DST-ICRL model can

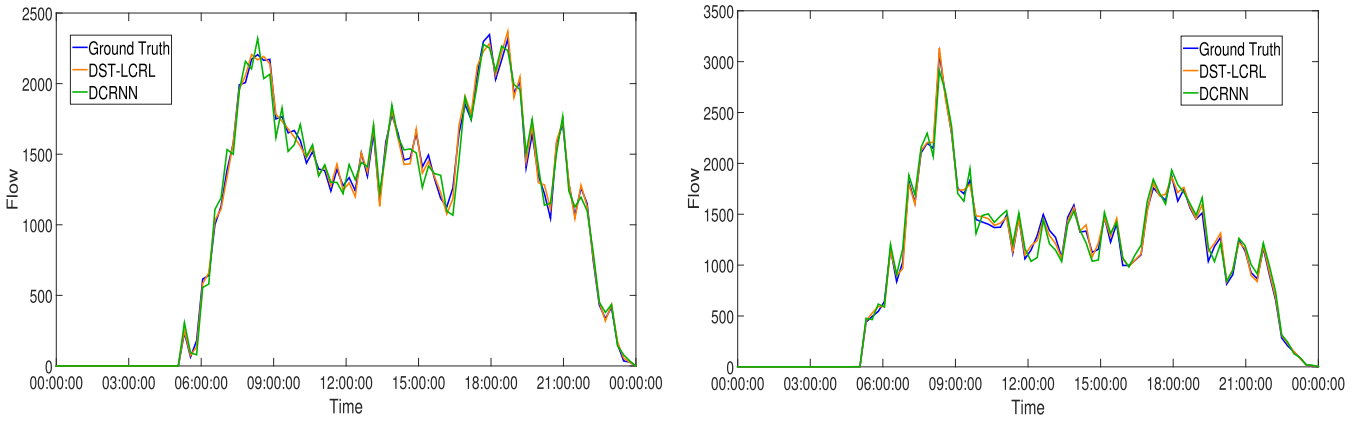


Fig. 8. Visualization of the prediction results of DST-LCRL and DCRNN in different hours of a day. (a) Beijing West Railway Station (Subway Station). (b) Xizhimen Subway Station.

reduce RMSE to 5.77 and MAE to 3.05, which are the best performances among all the methods, demonstrating that our proposed model has good generalization performance on other flow prediction tasks. Compared with the uniform sampling for training bike flows, the important sampling strategy can reduce the RMSE about by about 0.16. Compared with model DCRNN, which obtained the highest prediction accuracy in BikeNYC dataset among the baseline models, our DST-ICRL model decreases by 0.2 in RMSE and 0.8 in MAE. Compared with the taxi, the average travel distance of bike is relatively short, and the interaction among the bicycles is more independent. Thus overall the RMSE of the BikeNYC dataset is smaller than the TaxiBJ dataset.

TABLE VII
COMPARISON OF TRAINING TIME BASED ON GPU AND CPU. (TEST EVALUATIONS FOR ALL THE MODELS WERE PERFORMED BY CPUS)

Type	Datasets	Batch(s)	Train(h)	RMSE	MAE
CPU	SubwayBJ	1280	28	0.0007	0.0002
GPU	SubwayBJ	206	4.7	0.0007	0.0002
CPU	BusBJ	1620	36	0.00047	0.00014
GPU	BusBJ	275	4.3	0.00048	0.00014
CPU	TaxiBJ	280	5	14.07	4.62
GPU	TaxiBJ	34	0.8	14.11	4.64
CPU	BikeNYC	170	2	5.77	3.05
GPU	BikeNYC	28	0.34	5.77	3.05

H. Computational Efficiency Analysis

We also compare our models trained with different devices with different settings on DST-ICRL model, shown in Table VII It shows that GPU can speed up the training time by at least 6 times for the four datasets, while achieves comparable performance. Based on the same batch size, the train time of 1-Batch depends on the channel number and grid region size. Moreover, the 1-Batch training time for the BusBJ data is much more than others, because the channel number of the bus is larger than the subway, according to Table I. Note that the taxi and bike datasets are just 2-channel grids, and the grid size is smaller than the subway and bus, so the training times are smaller than the above's.

I. Visualization Analysis

To have a better understanding on the prediction performance of DST-ICRL, we visualize the forecasting results. We select the Beijing West Railway Station and Xizhimen Subway Station as case studies to show the prediction results. We show both the ground truth passenger flows and the predictions by DST-LCRL and the best baseline DCRNN in the two stations in different hours of a day in Figure 8(a) and Figure 8(b). The Beijing West Railway Station is located in the centre of Beijing, and there are a large number of passengers moving out and in this station each day. Xizhimen Subway Station is a busy transfer station with three traffic lines 2, 4 and 13. From the two figures one can have the following observations. First, compared with DCRNN, DST-ICLR can more accurately predict passenger flow peaks in rush hours of a day as well as multiple local peaks. This shows DST-ICLR is more robust than DCRNN. Second, DST-ICLR also performs better in predicting some sudden changes in the passenger flows than DCRNN. This is probably because DST-ICLR can better capture the spatial features of passenger flows in irregular neighborhood and multi-channel lines.

Next we visualize the predictions errors of the inflow and outflow of the Beijing subway passengers in the time slot from 8:00 am to 8:15 am on September 22, 2016 in Figure 9. Figure 9(a) and Figure 9(d) are the real passenger inflows and outflows, respectively. Figure 9(c) and Figure 9(f) show the visualization of the difference between the real flows and the predicted flows given by the DCRNN model. Figure 9(b) and Figure 9(e) show the visualization of the prediction errors of the two flows of DST-ICLR. One can have the following conclusions. (1) Compared with DCRNN, DST-ICLR can more accurately predict the passenger inflows and outflows. (2) Using multiple channels to represent the traffic flows in different subway lines is more suitable to

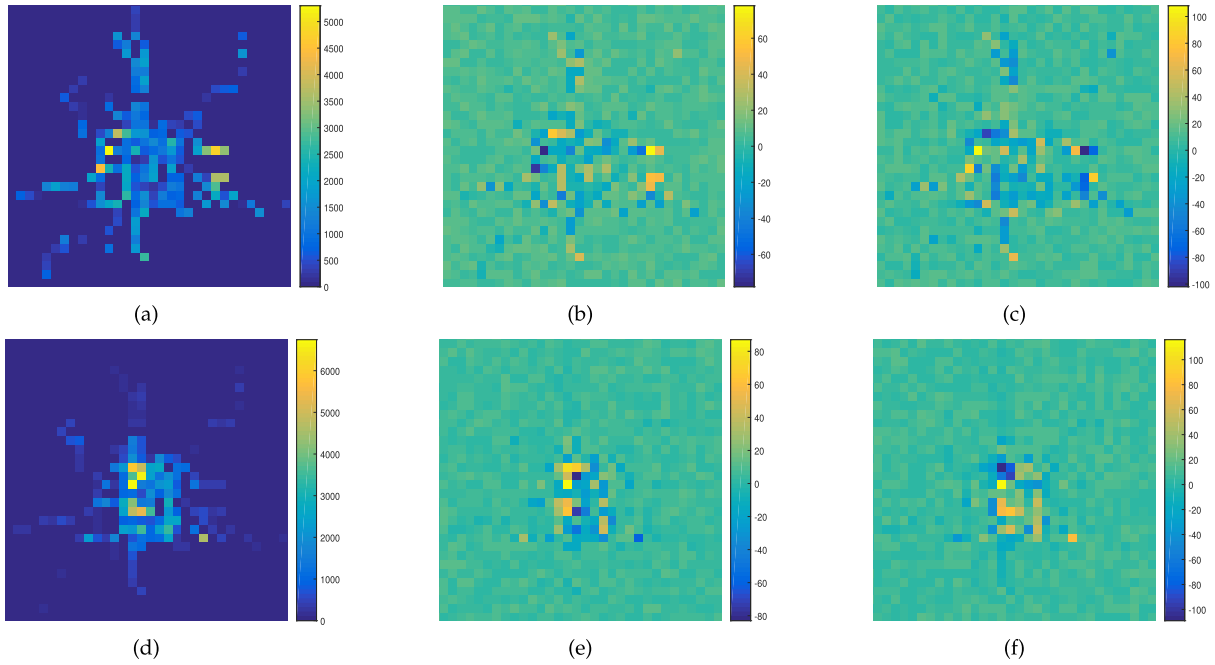


Fig. 9. Visualization of the real passenger flows and the prediction errors of DST-ICLR and DCRNN on the Beijing Subway dataset in the time slot 8:00-8:15 am on September 22, 2016. (a) Real Inflow. (b) Prediction error on inflow of DST-ICLR. (c) Prediction error on inflow of DCRNN. (d) Real Outflow. (e) Prediction error on outflow of DST-ICLR. (f) Prediction error on outflow of DCRNN.

achieve better prediction performance. (3) The forecasting accuracy on outflows is higher than that on inflows.

VI. Conclusions

In this paper, we propose a novel deep irregular convolutional residual LSTM model for forecasting the flows of crowds in transportation lines of a city. Our DST-ICRL model integrates multi-channel traffic representations, irregular convolution residual network to learn complex traffic spatial features, and use an importance sample strategy based LSTM units to learn temporal laws. We also take advantage of traffic passenger flows laws to a more abundant collection of recent traffic for accurate prediction. We evaluate our model on four types of crowd flows in Beijing and New York City, achieving performances which are significantly beyond 10 mainstream baseline methods, confirming that our model is better and more applicable to the crowd flow prediction tasks. In the future, we will consider incorporating other types of local external impacts such as traffic interchange, accidents and social events, and using hard attention model to improve flow prediction accuracy.

References

[1] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI*, 2017, pp. 1655–1661.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[3] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, 1984.

[4] T. J. Kim, L. L. Wiggins, and J. R. Wright, *Expert Systems: Applications to Urban Planning*. Springer, 2012.

[5] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, 2011.

[6] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, 2014, Art. no. 38.

[7] N. Raford and D. Ragland, "Pedestrian volume modeling for traffic safety and exposure analysis: The case of Boston, Massachusetts," *Safe Transp. Res. Educ. Center*, to be published.

[8] S. Manoharan, "Short term traffic flow prediction using deep learning approach," Ph.D. dissertation, School Comput., Nat. College Ireland, Dublin, Republic of Ireland, 2016.

[9] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[10] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, Apr. 2015.

[11] Y.-J. Wu, F. Chen, C.-T. Lu, and S. Yang, "Urban traffic flow prediction using a spatio-temporal random effects model," *J. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 282–293, 2016.

[12] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

[13] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[14] C. F. Daganzo and N. Geroliminis, "An analytical approximation for the macroscopic fundamental diagram of urban traffic," *Transp. Res. B, Methodol.*, vol. 42, no. 9, pp. 771–781, Nov. 2008.

[15] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, Sep. 2011, pp. 89–98.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[18] S. Zhang, G. Wu, J. P. Costeira, and J. M. F. Moura, "FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 3687–3696.

[19] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[20] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

[25] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[26] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI*, 2018, pp. 2588–2595.

[27] J. Ma, W. Wang, and L. Wang. (2017). "Irregular convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1706.07966

[28] J. Dai *et al.* (2017). "Deformable convolutional networks." [Online]. Available: https://arxiv.org/abs/1703.06211

[29] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw.*, Sep. 1999, pp. 850–855.

[30] B. Abdulhai, H. Porwal, and W. Recker, "Short-term traffic flow prediction using neuro-genetic algorithms," *J. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 3–41, 2002.

[31] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.

[32] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1644, no. 1, pp. 132–141, Jan. 1998.

[33] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flowas a seasonal ARIMA process: Theoretical basisand empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, 2003.

[34] Y. Kamarianakis and P. Prastacos, *Space-Time Modeling of Traffic Flow*. New York, NY, USA: Pergamon, 2005.

[35] S. Thajchayapong, J. A. Barria, and E. Garcia-Trevino, "Lane-level traffic estimations using microscopic traffic variables," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 1189–1194.

[36] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1310–1323, Jul. 2017.

[37] E. Mai and R. Hranac, "Twitter interactions as a data source for transportation incidents," in *Proc. Transp. Res. Board 92nd Annu. Meeting*, 2013, pp. 1613–1636.

[38] A. Schulz and P. Ristoski, "The car that hit the burning house: Understanding small scale incident related information in microblogs," in *Proc. 7th Int. AAAI Conf. Weblogs Social Media*, 2013, pp. 11–14.

[39] S. Wang *et al.*, "Computing urban traffic congestions by incorporating sparse gps probe data and social media data," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, 2017, Art. no. 40.

[40] S. Wang, L. He, L. Stenneth, P. S. Yu, Z. Li, and Z. Huang, "Estimating urban traffic congestions with multi-sourced data," in *Proc. 17th IEEE Int. Conf. Mobile Data Manage. (MDM)*vol. 1, Jun. 2016, pp. 82–91.

[41] T. H. Maze, M. Agarwal, and G. Burchett, "Whether weather matters to traffic demand, traffic safety, and traffic operations and flow," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1948, no. 1, pp. 170–176, 2006.

[42] J. Li *et al.*, "Graph CNNs for urban traffic passenger flows prediction," in *Proc. SmartWorld*, Oct. 2018, pp. 29–36.

[43] Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[45] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[47] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, p. 92.

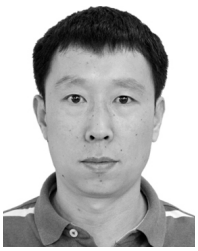
[48] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 736–744.

[49] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 685–693.

[50] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 265–272.

Bowen Du is currently an Assistant Professor with the State Key Laboratory of Software Development Environment, Beihang University, where he is also with the Beijing Advanced Innovation Center for Big Data and Brain Computing. His current research interests include data mining on intelligent transportation systems, smart city technology, and multi-source traffic data fusion.

Hao Peng is currently pursuing the Ph.D. degree with the State Key Laboratory of Software Development Environment, Beihang University, where he is also with the Beijing Advanced Innovation Center for Big Data and Brain Computing. His research interests include representation learning, urban computing, and text mining.



Senzhang Wang is currently an Associate Professor with the Collage of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing. His current research interests include data mining, urban computing, and social network analysis.

Md Zakirul Alam Bhuiyan is currently an Assistant Professor with the Department of Computer and Information Sciences, Fordham University. His research focuses on dependable cyber physical systems, WSN applications, network security, urban computing, and sensor-cloud computing.

Lihong Wang is currently a Professor with the National Computer Network Emergency Response Technical Team/Coordination Center of China. Her current research interests include information security, cloud computing, big data mining and analytics, information retrieval, and data mining.

Qiran Gong is currently pursuing the B.E. degree with the State Key Laboratory of Software Development Environment, Beihang University, Beijing, China.

Lin Liu is currently pursuing the Ph.D. degree with the State Key Laboratory of Software Development Environment, Beihang University, Beijing, China. Her research interests include urban computing and blockchain.

Jing Li is currently pursuing the Ms.D. degree with the State Key Laboratory of Software Development Environment, Beihang University, Beijing, China. His research interests include urban computing and big data computing.